(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
2 May 2002 (02.05.2002)

PCT

(10) International Publication Number
WO 02/35359 A2

(51) International Patent Classification[7]: G06F 12/00

(21) International Application Number: PCT/US01/42785

(22) International Filing Date: 26 October 2001 (26.10.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/266,286	26 October 2000 (26.10.2000)	US
60/278,469	23 March 2001 (23.03.2001)	US
60/278,408	23 March 2001 (23.03.2001)	US
60/278,409	23 March 2001 (23.03.2001)	US
60/278,285	23 March 2001 (23.03.2001)	US
09/681,644	15 May 2001 (15.05.2001)	US

(71) Applicant: **PRISMEDIA NETWORKS, INC.** [US/US]; 3080 North First Street, Second Floor, San Jose, CA 95134 (US).

(71) Applicants and
(72) Inventors: **HUANG, Tsan-Fung** [US/US]; 1817 Alpine Drive, San Marino, CA 91108 (US). **ISAACSON, Trygve** [US/US]; 901 William Drive, San Lorenzo, CA 94580 (US). **FLOOD, James, C., Jr.** [US/US]; 8540 SW Cashmur Lane, Portland, OR 97225 (US). **ORZEN, Matthew** [US/US]; 68 Whitney Street, San Francisco, CA 94131 (US).

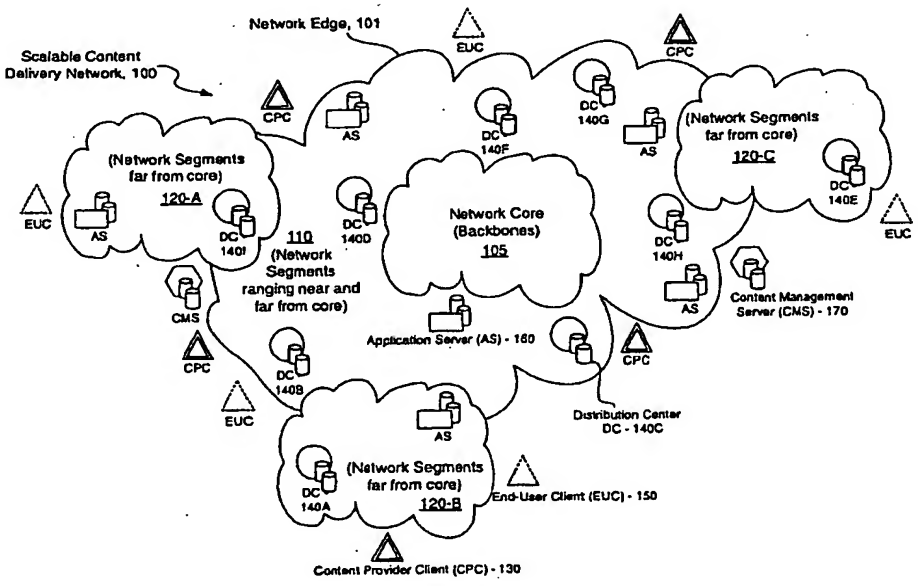
(72) Inventors: **SIM, Siew, Young**; 10435 Sterling Boulevard, Cupertino, CA 95014 (US). **CHAN, Desmond, Cho-Hung**; 55 Devonshire Avenue, Mountain View, CA 94043 (US). **CHAI, Wencheng**; 1067 Wunderlich Drive, San Jose, CA 95129 (US). **MILLS, George, Harlow**; 3215 Emerson Street, Palo Alto, CA 94580 (US).

(74) Agents: **RAY, Michael, B.** et al.; Sterne, Kessler, Goldstein & Fox P.L.L.C., Suite 600, 1100 New York Avenue, N.W., Washington, DC 20005-3934 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG,

[Continued on next page]

(54) Title: METHOD AND SYSTEM FOR MANAGING DISTRIBUTED CONTENT AND RELATED METADATA



(57) Abstract: The invention provides a method and system for creating an innovative file system that separates its directory presentation from its data store. The method and system include processing, division, distribution, managing, synchronizing, and reassembling of file system objects that does not delay the presentation of the content to the user, but also uses a reduced amount of storage space. The invention includes the ability to manage and control the integrity of the files distributed across the network, and the ability to serve and reconstruct files in real time using a Virtual File Control System.

WO 02/35359 A2

SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) **Designated States** *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Method and System For Managing Distributed Content and Related Metadata

Background of the Invention

Field of the Invention

The present invention relates to the storage and distribution of content over a network.

Related Art

Advances in telecommunications network communication and switching are moving ahead with great speed. However, distributing files between network locations can take significant amounts of time using conventional techniques. Transmission flow can be inconsistent. For example, when delivering large content, such as a media file of a movie, to a user, unacceptable delays in transmission can occur.

One conventional technique to avoid delay in presenting content to a user is to replicate copies of the content at various locations in the network. Such replication may reduce delay for a user near an available copy, but requires an inordinate amount of storage space. Management overhead is also increased. A management application is needed so that administrators and/or users can manage the replicated copies of content. Storage administrators must further be in constant alert because if any site runs out of storage, a new content replication will fail.

Other techniques include application level proxy caching, such as, web caching and streaming caching. Such caching does not require the deployment of unmanageable amount storage but only solves the problem for limited cases when content has already been cached in at the requesting locations. If a user request for a content that is not cached, the content has to be retrieved from the core, and the delay may be unacceptable. Another major limitation of a caching approach is that it is limited to one specific application.

What is needed is a method and system for storage and distribution of content over a network that can eliminate long haul transfer latency and does not require 100% replication to all locations. A method and system for storage and distribution of content is needed which provides intelligent storage management based on usage and location transparent access, and which is application agnostic, that is, is can be used with different types of applications.

Summary of the Invention

The invention overcomes the identified limitations and provides a method and system for creating an innovative file system that separates its directory presentation from its data store. The invention strikes an appropriate balance between the requirement of consistent speedy delivery and reducing storage requirements. The method and system includes division, distribution, and reassembling of files that does not delay the presentation of the content to the user, but also uses a reduced amount of storage space compared to conventional techniques. The method and system also includes the creation of an integrated read-write-able file system, and the distribution of file system objects that include volumes, directories, and files. The invention includes the ability to manage and control the integrity of the file system objects distributed across the networking using the metadata and meta file system architecture, and the ability to serve and reconstruct files in real time using a Virtual File Control System (VFCS) or VFCS cluster. The metadata and meta file system architecture provides means for distribution servers (DS) and VFCS to keep track of the distributed information on the network while VFCS conducts the organized reassembly of the information for delivery to the user.

An embodiment of the invention provides an improved mechanism for creating an integrated read-write-able file system for distributing large files throughout a computer network and delivering such files to end-user systems or an application servers. When the invention is implemented it provides

multiple users from many different locations a way to obtain access to file system objects without overburdening network resources. If, for example, a user wishes to download a large file, such as a video file, an embodiment of the invention provides a way to deliver that video file to the requesting user without straining the network. The system accomplishes this by breaking the file into multiple portions (segments or block files) and storing those portions in locations (e.g. nodes) distributed throughout the network. The present invention describes a technique to create a read-write-able integrated file system. It also describes a technique for breaking up the file and reconstructing it for distribution, as well as a technique to distribute file system objects.

An aspect of the invention is a method to create an integrated file system presentation based on a meta file system structure and the object metadata itself that supports the separation of the file system presentation and its data while allowing the data to be located throughout a network of nodes, and then reassembled in a timely fashion that is transparent to its users.

Another aspect of the invention is directed to dividing files into manageable, non-contiguous, file segments, re-arranging the file segments, and distributing these non-contiguous file segments for optimum network node performance. The non-contiguous file segments are reassembled for distribution to a client requesting the large payload file. The reassembly process is transparent to the user and provides the file data to the user with minimal latency.

Another aspect of the invention is a method to distribute, replicate, and synchronize file system objects among a network of nodes.

Another aspect of the invention is directed to serving non-contiguous file segments through the global file system presentation while presenting the original, unchunked content to users as a directory.

Another aspect of the invention is regarding the application of distribution and service policies to enable the guaranteed quality of service.

Another aspect of the invention is directed to how several VFCSs can be grouped as a virtual file system gateway cluster that increases I/O bandwidth while also providing load balancing and fault tolerance.

Advantages of the invention include the ability to store, track, distribute, and reassemble large payload files without delaying the presentation of content to the user, but also while requiring only a minimal amount of storage space.

The invention provides a method and apparatus for efficiently storing large files. A content network for delivering files to a user includes a plurality of storage elements disposed within a number of geographically distributed network nodes and configured to store portions of a file. A software management structure stores information regarding the content and location of each of the storage elements related to the file. A software content pruning structure is coupled to the software management structure and configured to selectively prune the content blocks stored in the storage elements to insure that the file is efficiently stored in the network.

In one or more embodiments, the portions and amount of a file maintained at each node depends on the available storage, popularity of the content, distribution criteria by the content owner, etc. Thus, least-likely to be used blocks of a file may be pruned (i.e., deleted from local storage) to make room for other highly desirable content. However, although the least likely to be used blocks of a file are pruned, the entire content of a file may be maintained at a node in the scalable content delivery network, so long as the content owner wants the content to remain in the network. In this way, large files can be stored efficiently.

Further features and advantages of the present invention, as well as the structure and operation of various embodiments of the present invention, are described in detail below with reference to the accompanying drawings.

Brief Description of the Drawings

The accompanying drawings, which are incorporated herein and form part of the specification, illustrate the present invention and, together with the description, further serve to explain the principles of the invention and to enable a person skilled in the pertinent art to make and use the invention. In the accompanying drawings:

Figure 1 is an illustration of a scalable content delivery network for delivering file system objects according to an embodiment of the present invention;

Figure 2 is an illustration of a virtual tree arrangement of the nodes for control information communication in accordance with an embodiment of the present invention;

Figure 3 is an illustration of the attribute bitmap and rolled up bitmap, in accordance with an embodiment of the present invention;

Figures 4A-4C are the simplified layouts of a distribution center in accordance with embodiments of the present invention;

Figures 5A-5C provide three illustrative embodiments of the application server cluster in accordance with the present invention;

Figure 6 presents a layout of a Virtual File Control System cluster in accordance with an embodiment of the present invention;

Figures 7A-B shows the process of introducing a new file system object into a SCDN, or updating or deleting an existing file system object from a SCDN in accordance with an embodiment of the present invention;

Figure 7C shows the application of policies for quality of service based on file system object and object type in accordance with an embodiment of the present invention;

Figure 8 is an illustration of linear and non-linear file structures as used in the present invention;

Figure 9 shows the process of decomposing a file into block files for storage in accordance with an embodiment of the present invention;

Figures 10A-B are two illustrations of decomposed file in accordance with an embodiment of the present invention;

Figures 11A-B are illustrative embodiments of the distribution of a file system object and metadata within the network of the present invention;

Figure 11C illustrates how distribution servers work together to distribute and replicate meta information and content dynamically so that each server presents a global file system view that is an aggregated view of the entire network.

Figures 12A-C are illustrative embodiments of the meta file system structure, block file structure of an underlying file system, and metadata examples in accordance with the present invention;

Figures 13A-C are illustrative embodiments of the volume, directory, and file metadata in accordance with the present invention;

Figure 13D is an illustrative embodiment of the block index array metadata in accordance with the present invention;

Figure 14A is a diagram showing the process of reconstructing a file from one or multiple block files in accordance with the present invention;

Figure 14B is a diagram showing the algorithm for locating data in the process of reconstructing a file in real time in accordance with the present invention;

Figures 15A-C are three illustrative embodiments of a VFCS in accordance with the present invention;

Figure 16 is a flow diagram of the operations of a VFCS server performed during the VFCS initialization process to create a global file system presentation in accordance with an embodiment of the present invention;

Figure 17A shows the VFCS server operations performed during run time in accordance with an embodiment of the present invention;

Figure 17B shows the application of policies for quality of service based on the user and file system object type in accordance with the present invention;

Figure 18 is an illustration of the VFCS modules in accordance with an embodiment of the present invention;

Figure 19 is a flow diagram of the operations of a VFCS server handling of a read request in accordance with an embodiment of the present invention;

Figure 20 is a diagram illustrating a server request distribution capability of an SCDN load balancer in accordance with an embodiment of the current invention;

Figure 21 is a diagram illustrating a server redundancy function provided by an SCDN load balancer in accordance with an embodiment of the current invention;

Figure 22 is a diagram illustrating the instant fail-over capability of an SCDN load balancer as a stateless load balancer in accordance with an embodiment of the current invention;

Figure 23 is a diagram showing an SCDN load balancer redirecting a packet in accordance with an embodiment of the present invention;

Figure 24 is an illustrative embodiment of an SCDN load balancer redirecting packets with direct server return in accordance with an embodiment of the present invention;

Figure 25 is a flow diagram of an inbound packet redirection process performed by an SCDN load balancer in accordance with an embodiment of the present invention;

Figure 26 is a flow diagram of a health check process as performed by an SCDN load balancer in accordance with an embodiment of the present invention;

Figure 27 is an illustration of a station showing an exemplary a control unit and data repositories;

Figure 28 is a state diagram showing the storage management steps performed by one embodiment of the present invention;

Figures 29A-E break the operations of Figure 28 down into smaller subtasks;

Figure 30 illustrates the Storage Management knowledge base tables; and

Figure 31 is an example computer system and computer program product in which the present invention is implemented primarily in software.

Detailed Description of the Preferred Embodiments

The following description is for the best modes presently contemplated for practicing the invention. This description is not to be taken in a limiting sense, but is made merely for the purpose of describing the general principles of the invention. The scope of the invention should be ascertained with reference to the claims.

The present invention is related to a method and system for storing and distributing content. In particular, the invention provides a highly efficient architecture and technique for processing, storing and serving content to a user for education, entertainment, business, or any other purpose. A method and system according to an embodiment of the present invention creates an advanced read-write-able integrated network file system in which directory presentation and data store are separated. The method and system includes division, distribution, and re-assembling of files that does not delay the presentation of the content to the user, but also does not require an inordinate amount of storage space. The method and system also includes creation of an integrated file system, and distribution of file system objects including volumes, directories, and files.

The invention is described with reference to specific architectures and protocols. Those skilled in the art will recognize that the description is for

illustration and to provide the best mode of practicing the invention. One embodiment of the invention provides an improved mechanism for dividing and distributing files (referred to as payload or content) throughout a computer network. Another embodiment of the invention provides a method to create an integrated file system view of multiple content nodes. Another embodiment of the invention provides a method to distribute, replicate, and synchronize the update of file system objects such as volumes, directories, and files. In the following description, numerous specific details are set forth to provide a more thorough description of embodiments of the invention. The description is not meant to be limiting. For example, reference is made to Internet Protocol and UNIX, but any packet protocol may be used and any operating system may be used.

When the invention is implemented in accordance with an embodiment of the invention it provides end-user systems with a way to access file system objects without overburdening the network utilized by the end-user system to transmit data. In one embodiment of the invention, the system accomplishes this by breaking the file into multiple portions (segments or tracks) and storing those portions and other file system objects in locations (e.g., nodes) distributed throughout the network. The portions and other file system objects stored throughout the network are distributed utilizing a flow optimization technique that provides for the intelligent management of the all file system objects and portions of data. Thus, file system objects and portions of the file are stored in locations that minimize the amount of time it takes to deliver the portion to the end-user system. These locations minimize the latency associated with delivering the data to the end-user system and are referred to herein as the edge of the network.

Each node at the edge of the network embodying aspects of the invention is configured to appear as if it has the file stored locally when portions of the file are really stored on other nodes located throughout the network. This greatly increases the virtual storage capacity of each network

node without consuming system resources. The nodes distribute and replicate data blocks and other file system objects in a manner that maximizes data transfer efficiency while minimizing bandwidth consumption. When the end-user system issues a request for content (e.g., a file) the request is routed to the nearest node and the node imports non-resident data of the requested content from other nodes in a manner that requires the least time and cost. The end result is that each network node has access to numerous or all file system objects (volumes, directories, and files) without having to store and maintain the full content of each of those objects locally.

One or more embodiments of the present invention provide efficient methods and systems for dividing a file for storage and reconstructing the file for delivery. The process of dividing a large payload file content is called chunking and is described in detail below. Another embodiment of the present invention provides a method to create an integrated file system from multiple nodes. Another embodiment of the present invention provides a method to distribute, replicate, and synchronize file system objects among a network of nodes. Another embodiment of the present invention provides a method and system for clustering a group of virtual file systems. This clustering of a group of virtual file systems increases reliability and availability and at the same time increases I/O bandwidth by load balancing. These embodiments are described in more detail below.

A. Network Architecture

1. Scalable Content Delivery Network

Figure 1 provides a view of a scalable content delivery network (SCDN) 100 for delivering large payloads according to an embodiment of the present invention. SCDN 100 may be a network such as the Internet that conceptually includes a network core 105 (i.e., the backbone), intermediate

network segments 110 ranging "near" and "far" from the core, and network segments "far" from core 120-A through 120-C (collectively 520). "Near" and "far" relate to distance and are intended to indicate relative path latencies (short or long, respectively) to the core, such latencies generally depend on the number of intermediate hubs (e.g., switches, routers, and the like) that are traversed to reach the high-speed backbones that form the core of the network and through which much of the network traffic is routed. Note that each intermediate hub may perform some limited processing, which adds latency, before forwarding the traffic to the next hub.

Figure 1 shows a plurality of Content Provider Clients (CPCs) 130, a plurality of End-User Clients (EUCs) 150, and one or more Content Management Servers (CMSs) 170, all located beyond network edge 101. This is arrangement is illustrative and not intended to be limiting. For example, a CPC 130, EUC 150, and/or CMS 170 can be located anywhere in a network including beyond a network edge, at a network edge, or at any location within a network such as within a network segment or core.

In general, the content provider client 130 may be connected (or assigned) to a content management server 170, which in turn is connected to its assigned distribution center 140, or content provider client 130 may be connected (or assigned) to any distribution center 140. In this environment, any connection supported by the SCDN 100 can be used. Examples of such connections include, but are not limited to, a physical link (over any medium wired or wireless), data link, logical link, permanent virtual circuit, switched virtual circuit, connection-oriented protocol, connectionless protocol, or any other direct or indirect network connection and/or protocol and combinations thereof.

A content provider client may be an application for managing contents in the network, or it may be a general file system client that connects to a Virtual File Control System (not shown) in a distribution center 140. A content owner creates, renames, moves, deletes, and manages volumes and

directories through a respective CPC 130. A content owner also uploads, reads, updates, and manages files in the SCDN 100 through his or her CPC 130. EUC 150 provides an end-user of the content access to files in SCDN 100. For example, EUC 150 may be any kind of browser (including but not limited to a web browser or any file system browser) running on an end-user's local device. Any type of end user device that can support an end-user client 150 can be used including, but not limited to, a computer (e.g., a personal computer, workstation, or server), set-top box, television set, telephone, or a hand-held computing device (e.g., organizers, palm-top devices).

Network edge 101 may be far from network core 105. However, the distance (i.e., path latency) between the core and the edge may not be uniform, and may vary considerably for a given CPC or EUC. One embodiment of the present invention places a plurality of Distribution Centers (DC) 140A-140I for maintaining payloads at the edge of the network thereby reducing or eliminating latency for respective end user clients 150. Payload content from a content owner is pushed from one distribution center to other distribution centers at the edge of the network. An end-user seeking access to particular payload content is serviced (via a network file system client or an application server) from the nearest distribution center containing the desired content. Latency due to path considerations is minimized since content is distributed to the end-user (e.g., to a respective EUC 150) via a plurality of application aervers (AS) 160 and distribution centers 140 located at network edge 101. Thus, distribution involves obtaining any file system objects from a content provider and geographically placing these objects or portions of each objects at the distribution centers which are generally located close to the edge of the network.

The distribution centers 140A-140I in SCDN 100 of Figure 1 are virtually arranged in the form of a tree 200 as illustrated in Figure 2, for example. This virtual tree arrangement is primarily used for communication of control information and signals amongst the nodes of scalable content

delivery network 100. Data downloads can be performed from any node in the network having the desired data, preferably the nearest node (network-distance-wise). Nodes A through I of Figure 2 represent DCs 140A through 140I, respectively. The nodes are arranged in a logical order. For example, assuming node B represents Europe-England, then logical child nodes in Europe might be Europe-France (e.g., node D) and Europe-Germany (e.g., node E), and a child node of Europe-France might be Europe-Italy (e.g., node H). In this example where the left side of the tree represents Europe, the right side may represent Asia.

Node A is the root node and may represent a central control station, for example. In one or more embodiments, each node A-I in tree 200 has a unique attribute set representing the name of the node. The attribute set for a node is stored at a respective DC 140A-140I and can be represented in any convenient data structure. For example, the attribute set can be represented as a variable bitmap (a bitmap is the binary representation of an object, e.g., a number). Each node also contains a representation of the attribute set of each of the node's children, grandchildren, great grandchildren, etc. (i.e., all nodes emanating from that node as a root node – lineal descendants). This representation is called the "Rolled Up Set of Attributes" and any convenient data structure can be used for it. Thus the rolled up attribute of a node is the representation of the rolled up attribute of its children. For example, a "Rolled Up Bitmap", which is a combination of the rolled up attribute bitmaps of all the node's children, may be used. A "Rolled Up Bitmap" may be defined as the "binary OR" (also called a "Bitwise OR") of the rolled up attributes of the node's children.

Figure 3 is an illustration of example attribute bitmaps 300, 310, 320, 330 and rolled up bitmaps 340, 350 in accordance with an embodiment of the present invention. Each bitmap 300-350 uses 16 bits for illustration purposes, but since the bitmaps are variable, they may vary as needed to identify each node and provide other information.

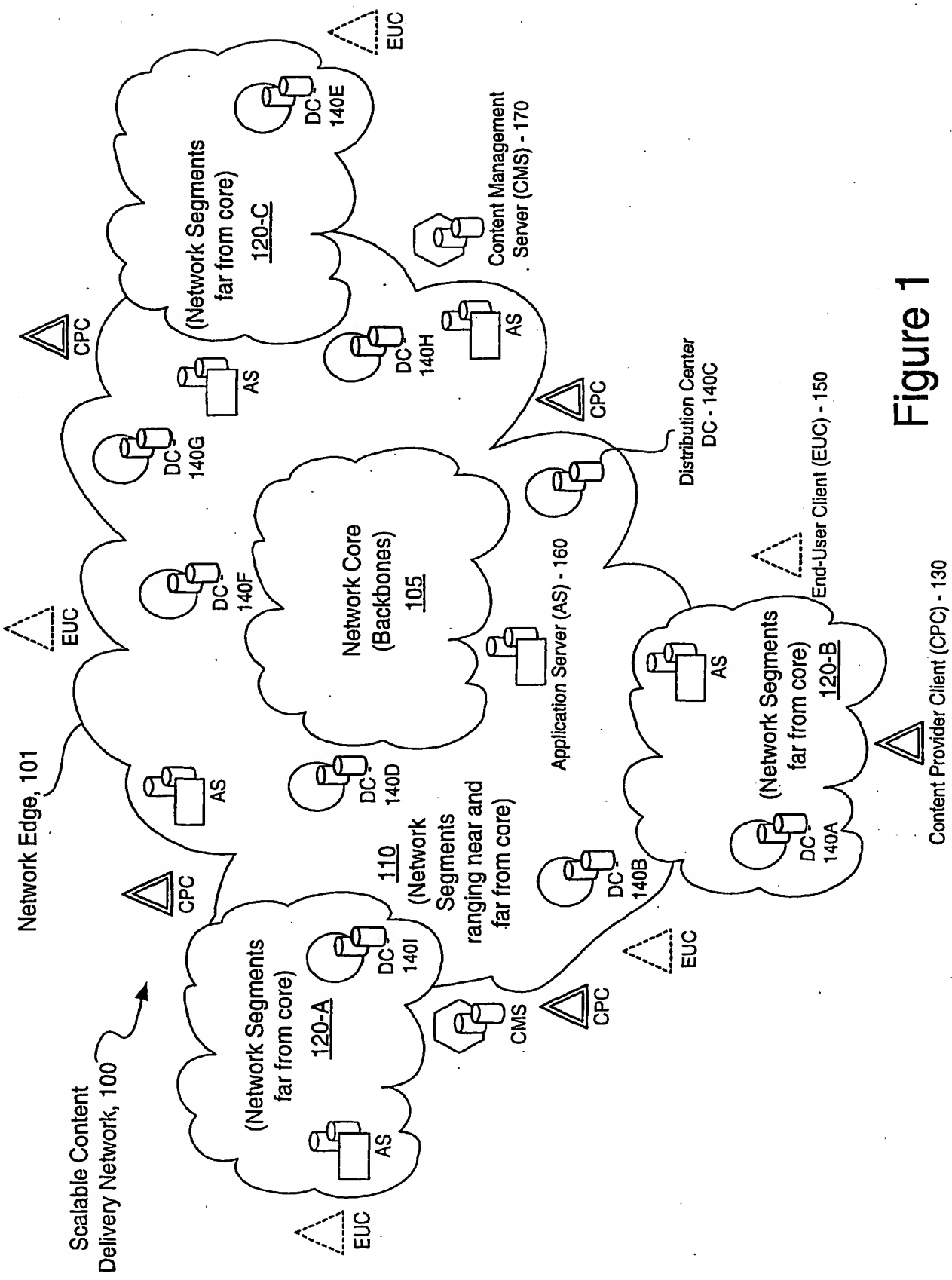

Scalable Content Delivery Network, 100
Network Edge, 101
(Network Segments far from core) 120-A
(Network Segments ranging near and far from core) 110
Network Core (Backbones) 105
(Network Segments far from core) 120-B
(Network Segments far from core) 120-C
Content Management Server (CMS) - 170
Application Server (AS) - 160
Distribution Center DC - 140C
End-User Client (EUC) - 150
Content Provider Client (CPC) - 130
AS
CMS
CPC
EUC
DC 140A
DC 140B
DC 140D
DC 140E
DC 140F
DC 140G
DC 140H
DC 140I

Figure 1

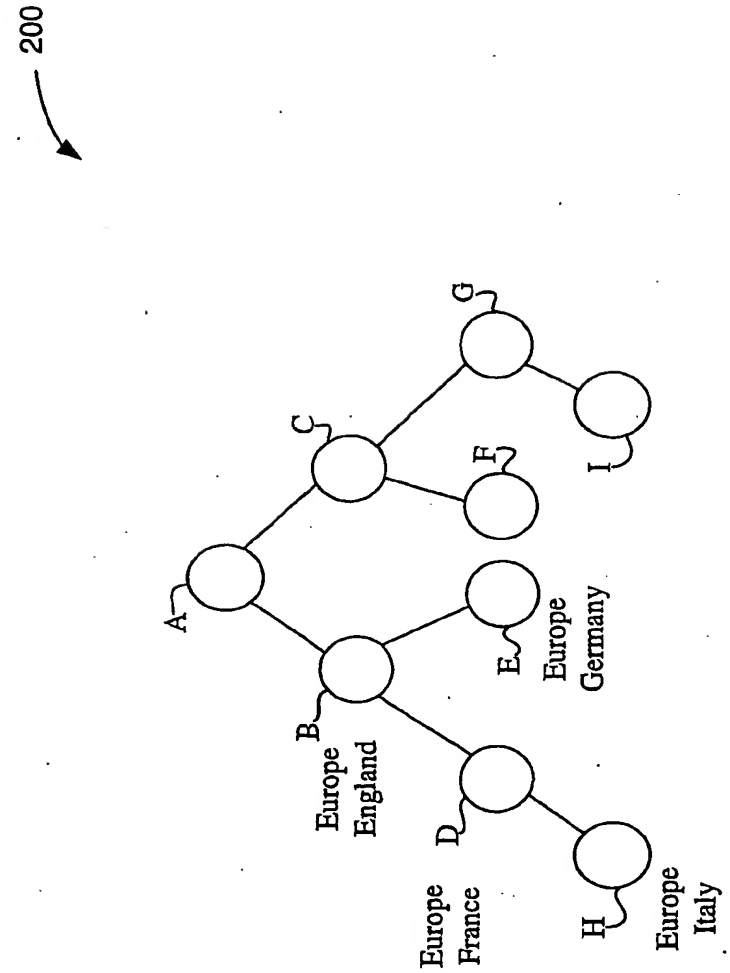

200
A
B
Europe
England
C
D
Europe
France
E
Europe
Germany
F
G
H
Europe
Italy
I

Figure 2

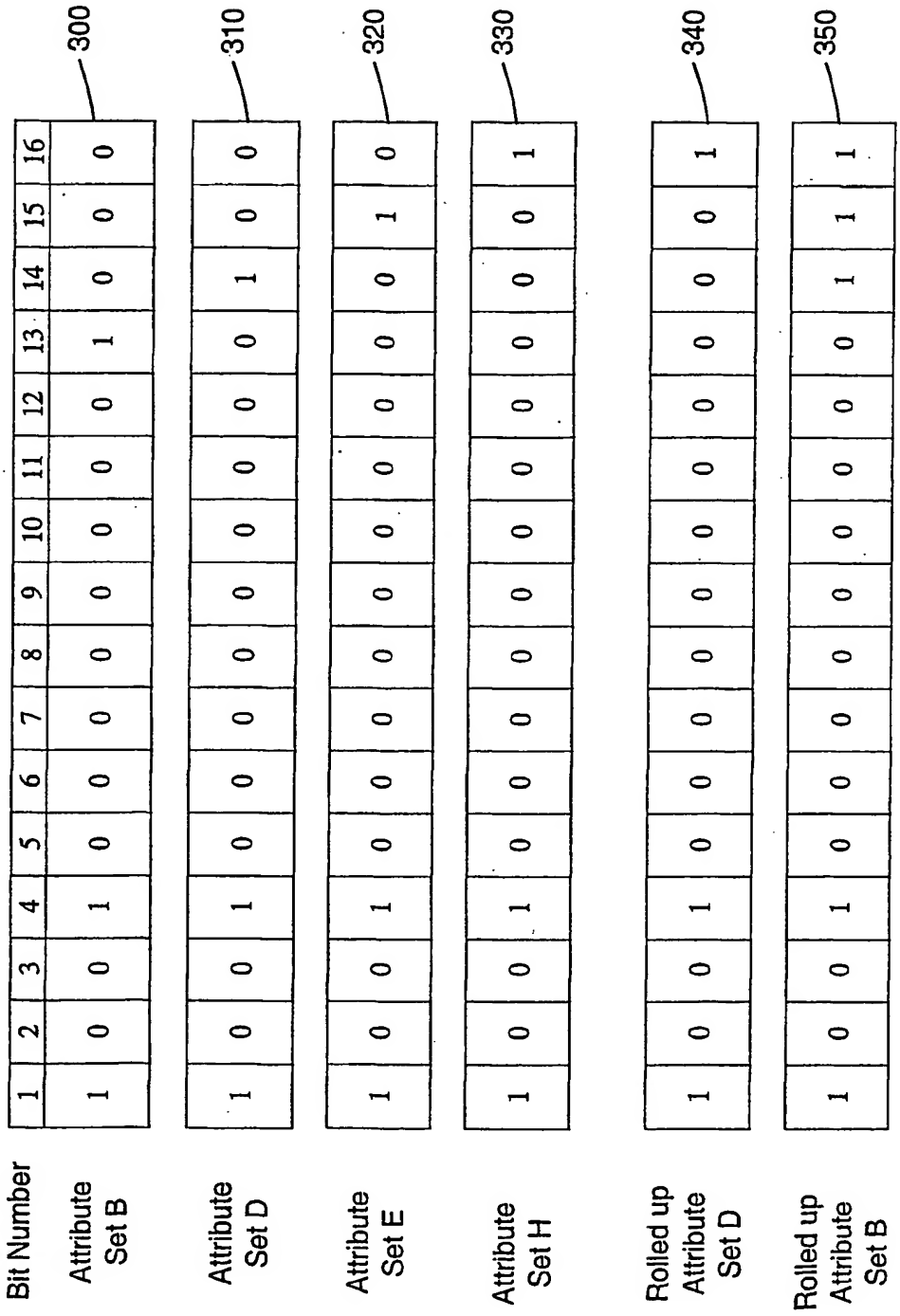

Bit Number
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Attribute Set B
1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0
300
Attribute Set D
1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0
310
Attribute Set E
1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0
320
Attribute Set H
1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1
330
Rolled up Attribute Set D
1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1
340
Rolled up Attribute Set B
1 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1
350

Figure 3

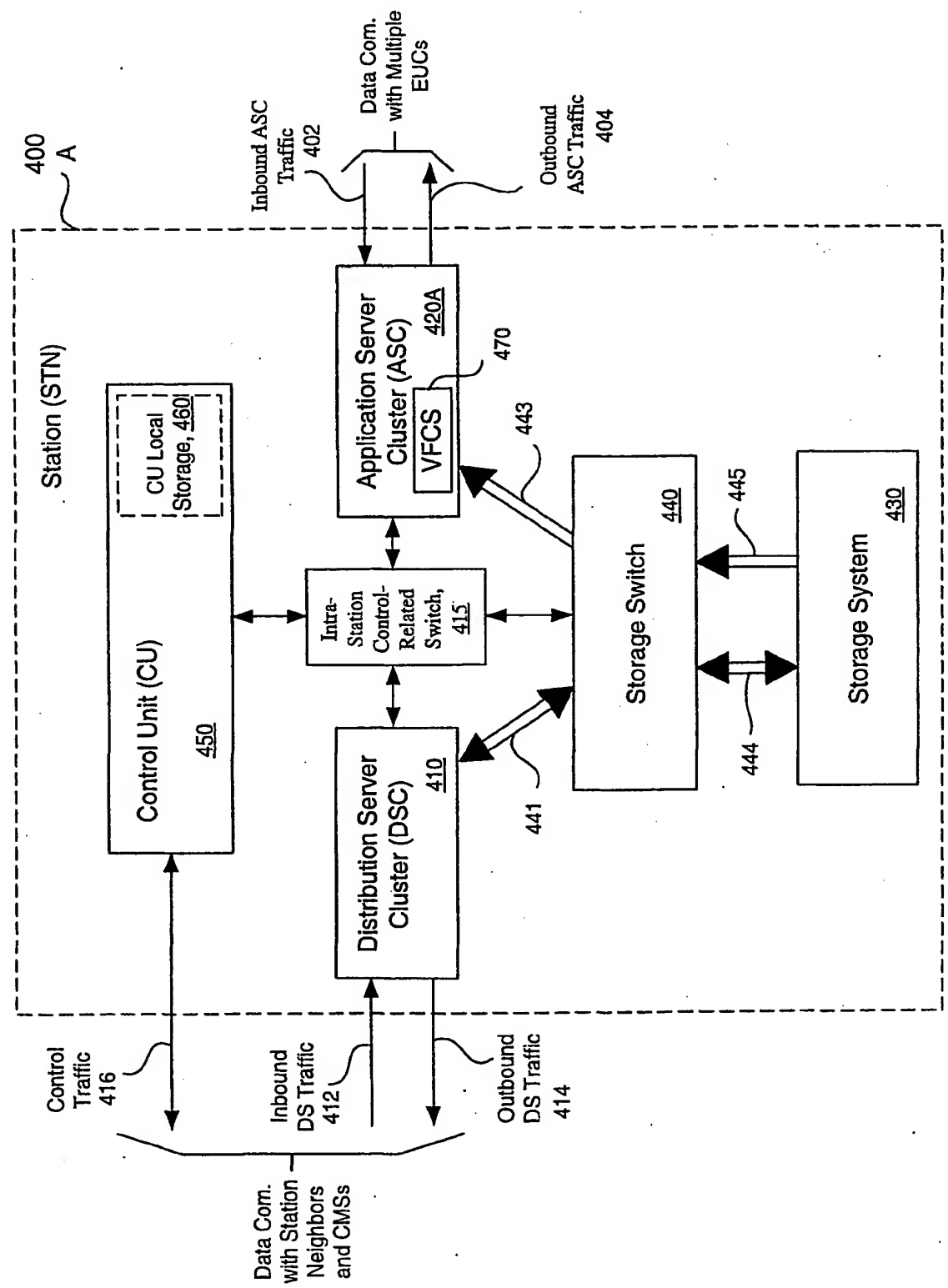

400
A
Station (STN)
Inbound ASC Traffic 402
Data Com. with Multiple EUCs
Outbound ASC Traffic 404
Application Server Cluster (ASC) 420A
VFCS
470
443
CU Local Storage, 460
Control Unit (CU) 450
Intra-Station Control-Related Switch, 415
Storage Switch 440
445
Storage System 430
Distribution Server Cluster (DSC) 410
441
444
Control Traffic 416
Inbound DS Traffic 412
Outbound DS Traffic 414
Data Com. with Station Neighbors and CMSs

Figure 4A

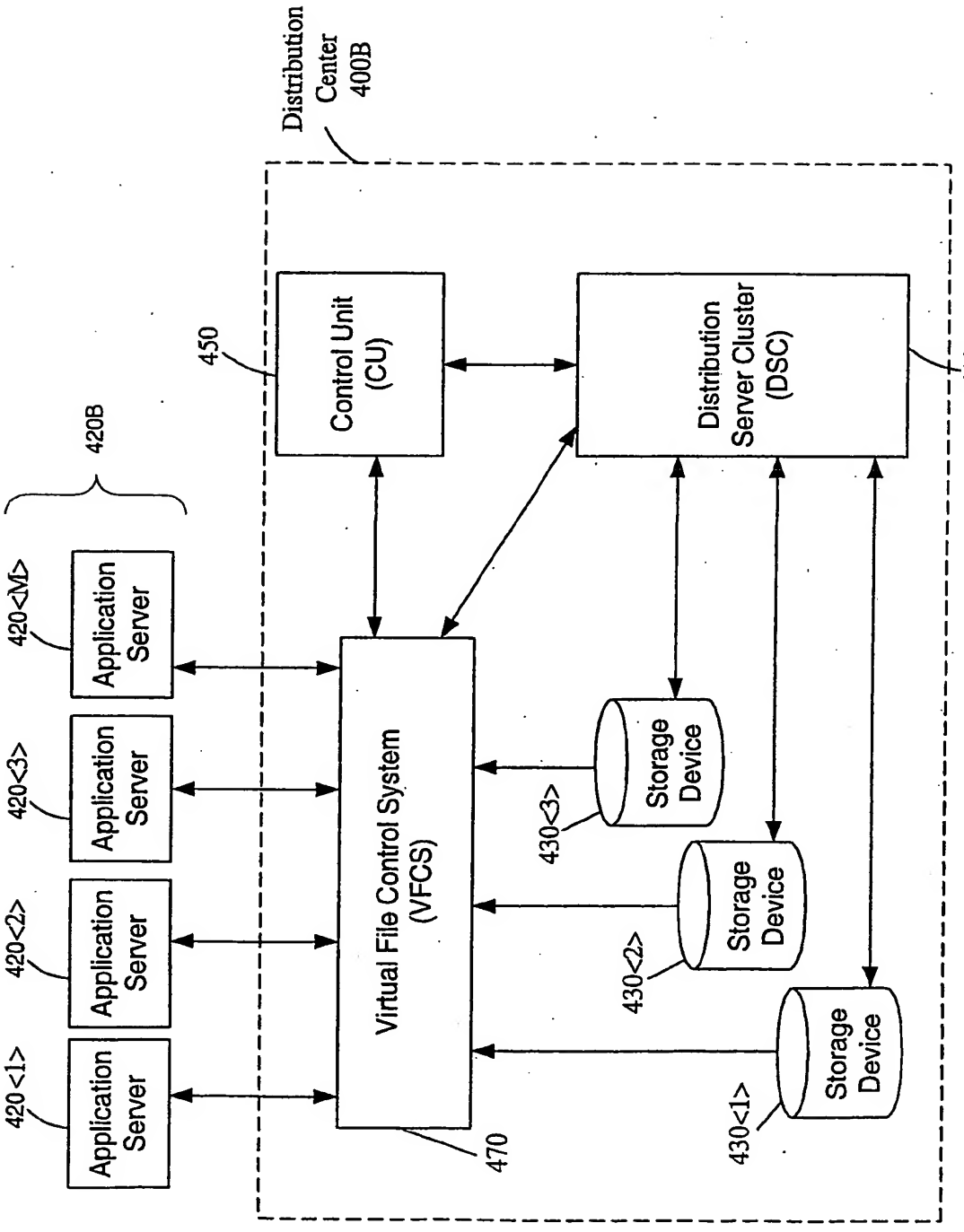

Distribution Center 400B
Control Unit (CU)
450
Distribution Server Cluster (DSC)
410
420B
Application Server
420<M>
Application Server
420<3>
Application Server
420<2>
Application Server
420<1>
Virtual File Control System (VFCS)
470
Storage Device
430<3>
Storage Device
430<2>
Storage Device
430<1>

Figure 4B

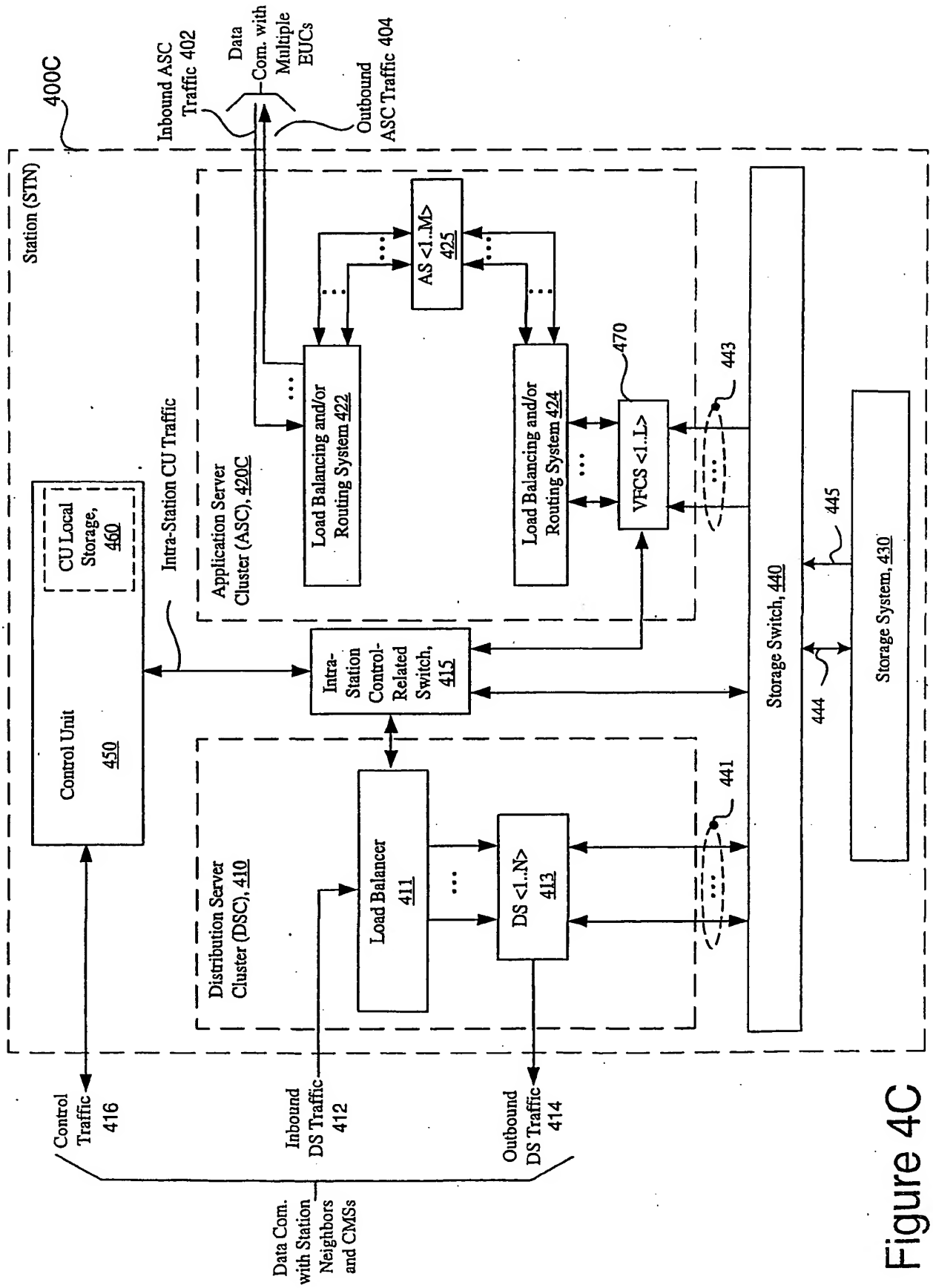

400C
Station (STN)
Inbound ASC Traffic 402
Data Com. with Multiple EUCs
Outbound ASC Traffic 404
Application Server Cluster (ASC), 420C
Load Balancing and/or Routing System 422
AS <1..M> 425
Load Balancing and/or Routing System 424
VFCS <1..L>
470
443
445
Storage Switch, 440
Storage System, 430
444
CU Local Storage, 460
Control Unit 450
Intra-Station CU Traffic
Intra-Station Control-Related Switch, 415
Distribution Server Cluster (DSC), 410
Load Balancer 411
DS <1..N> 413
441
Control Traffic 416
Inbound DS Traffic 412
Outbound DS Traffic 414
Data Com. with Station Neighbors and CMSs

Figure 4C

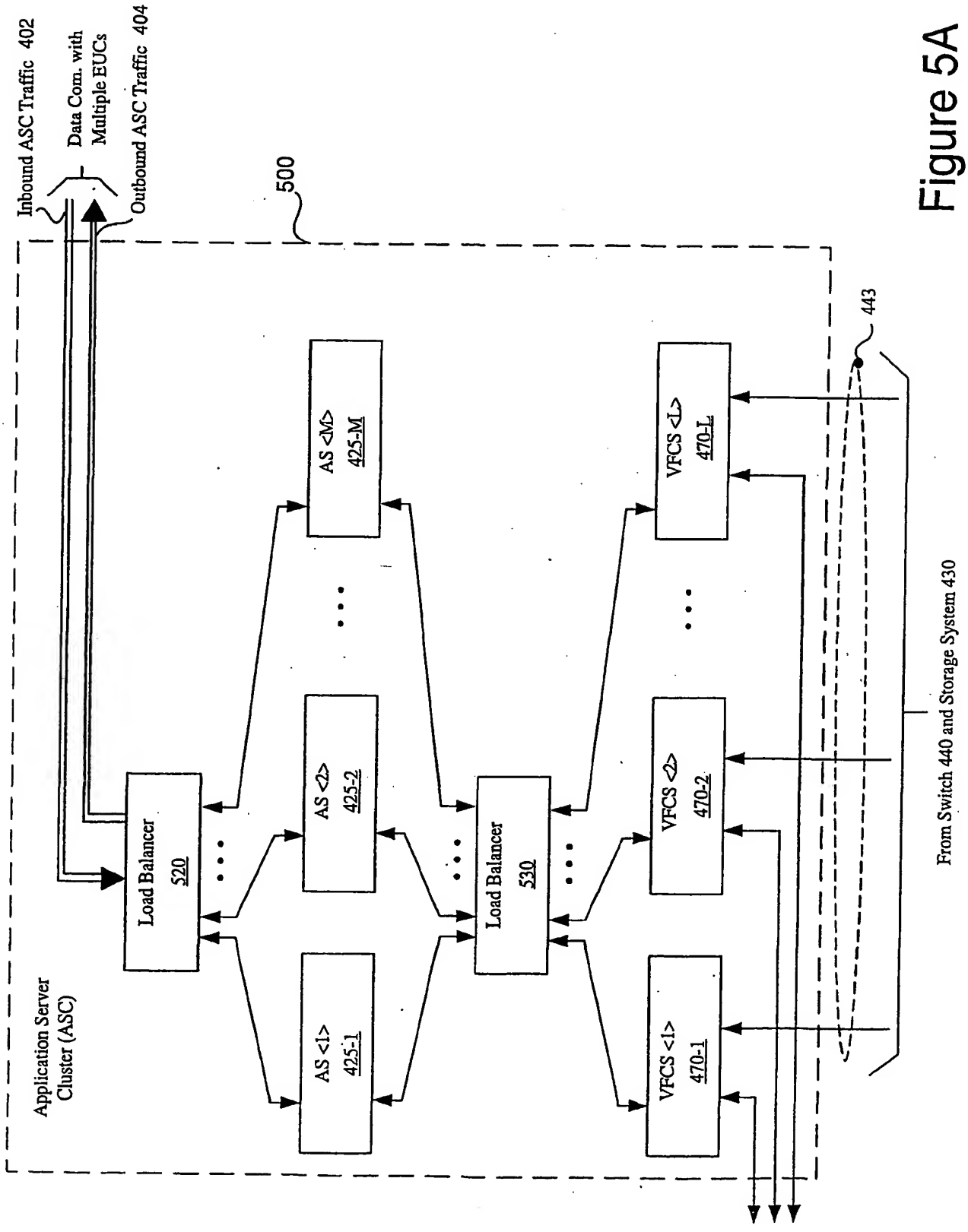

Inbound ASC Traffic 402
Data Com. with Multiple EUCs
Outbound ASC Traffic 404
500
Application Server Cluster (ASC)
Load Balancer 520
Load Balancer 530
AS <1> 425-1
AS <2> 425-2
AS <M> 425-M
VFCS <1> 470-1
VFCS <2> 470-2
VFCS <L> 470-L
443
From Switch 440 and Storage System 430

Figure 5A

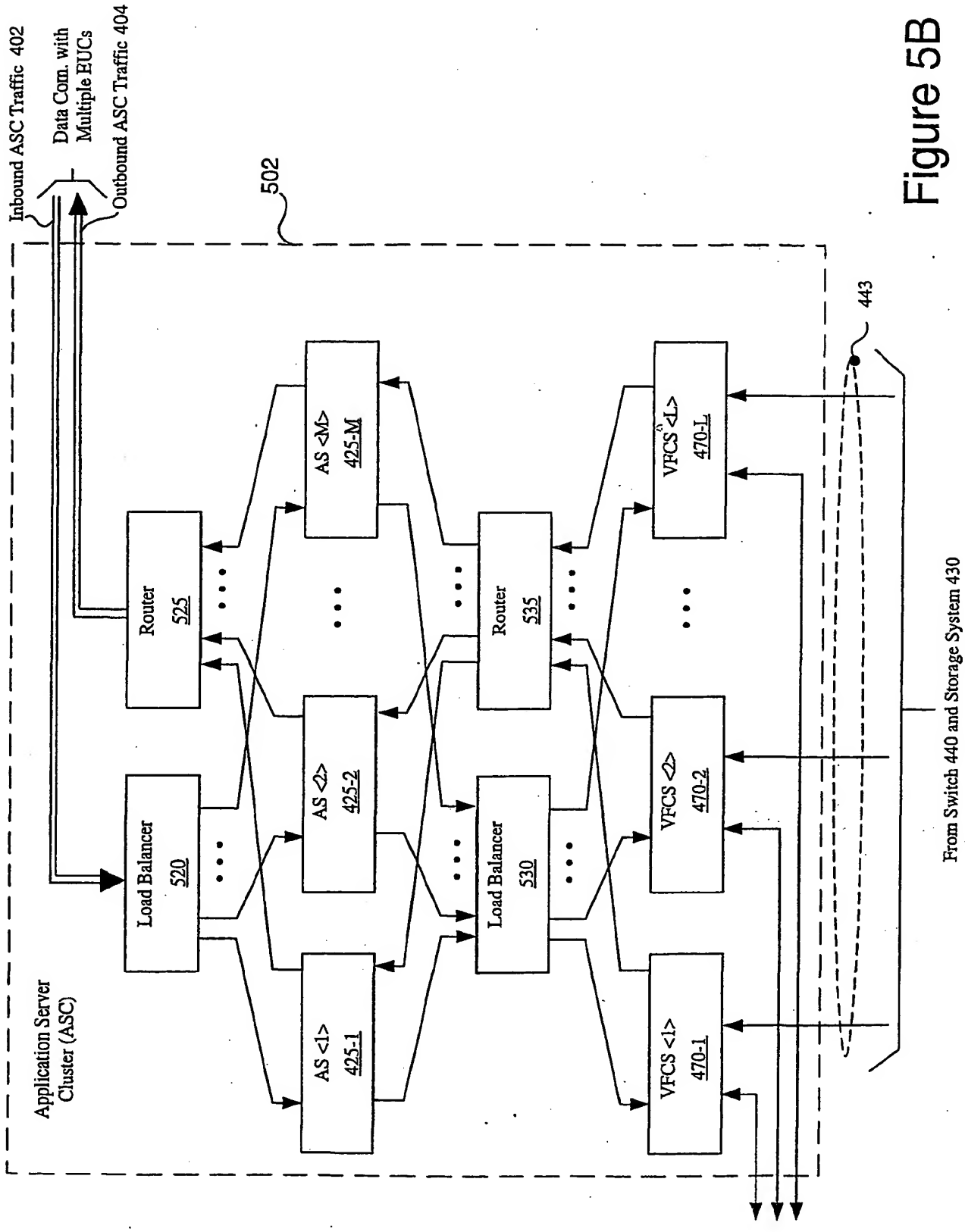

Inbound ASC Traffic 402
Data Com. with Multiple EUCs
Outbound ASC Traffic 404
502
Application Server Cluster (ASC)
Load Balancer 520
Router 525
AS <1> 425-1
AS <2> 425-2
AS <M> 425-M
Load Balancer 530
Router 535
VFCS <1> 470-1
VFCS <2> 470-2
VFCS <L> 470-L
443
From Switch 440 and Storage System 430

Figure 5B

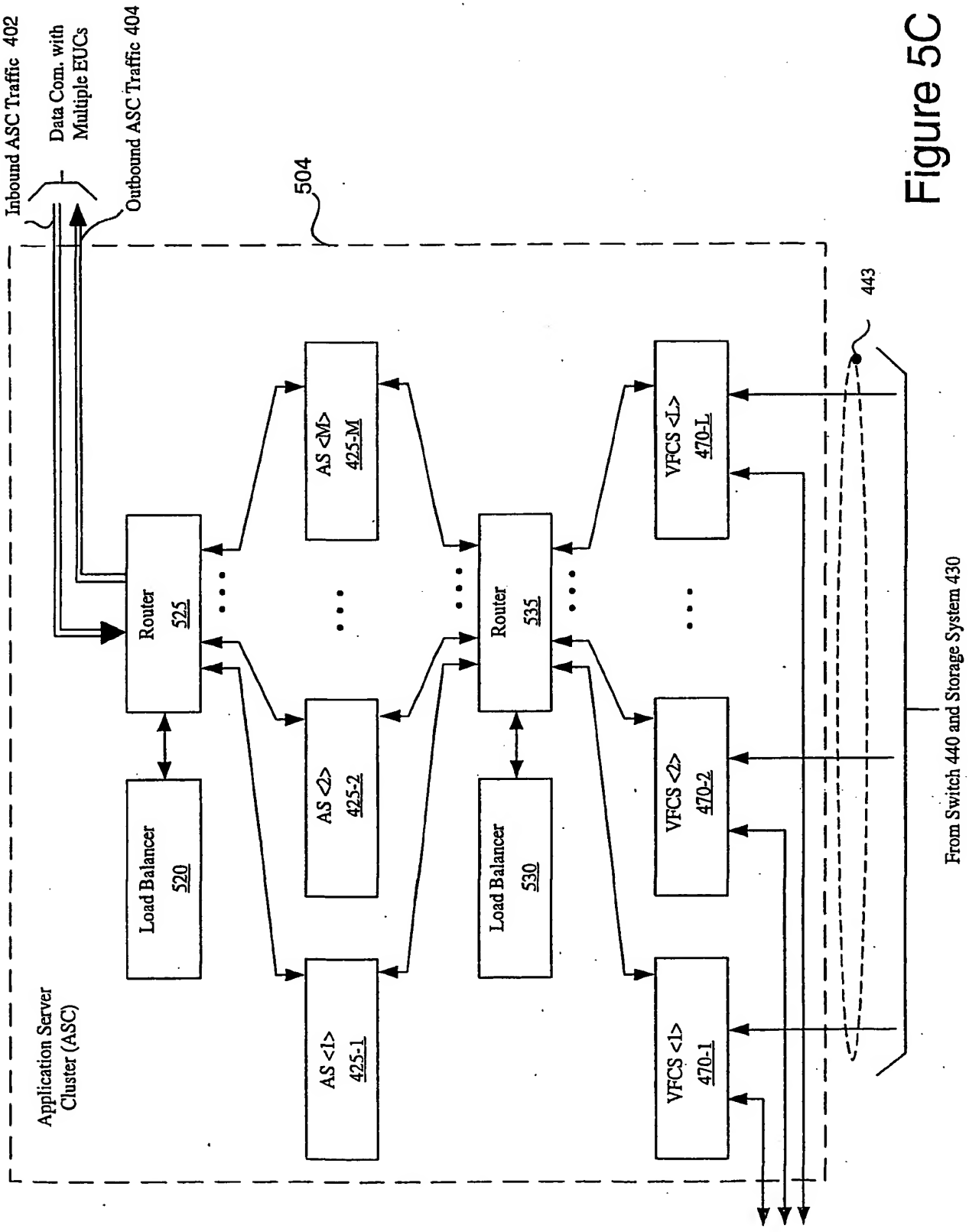

Inbound ASC Traffic 402
Data Com. with Multiple EUCs
Outbound ASC Traffic 404
504
443
Router 525
Load Balancer 520
AS <1> 425-1
AS <2> 425-2
AS <M> 425-M
Router 535
Load Balancer 530
VFCS <1> 470-1
VFCS <2> 470-2
VFCS <L> 470-L
Application Server Cluster (ASC)
From Switch 440 and Storage System 430

Figure 5C

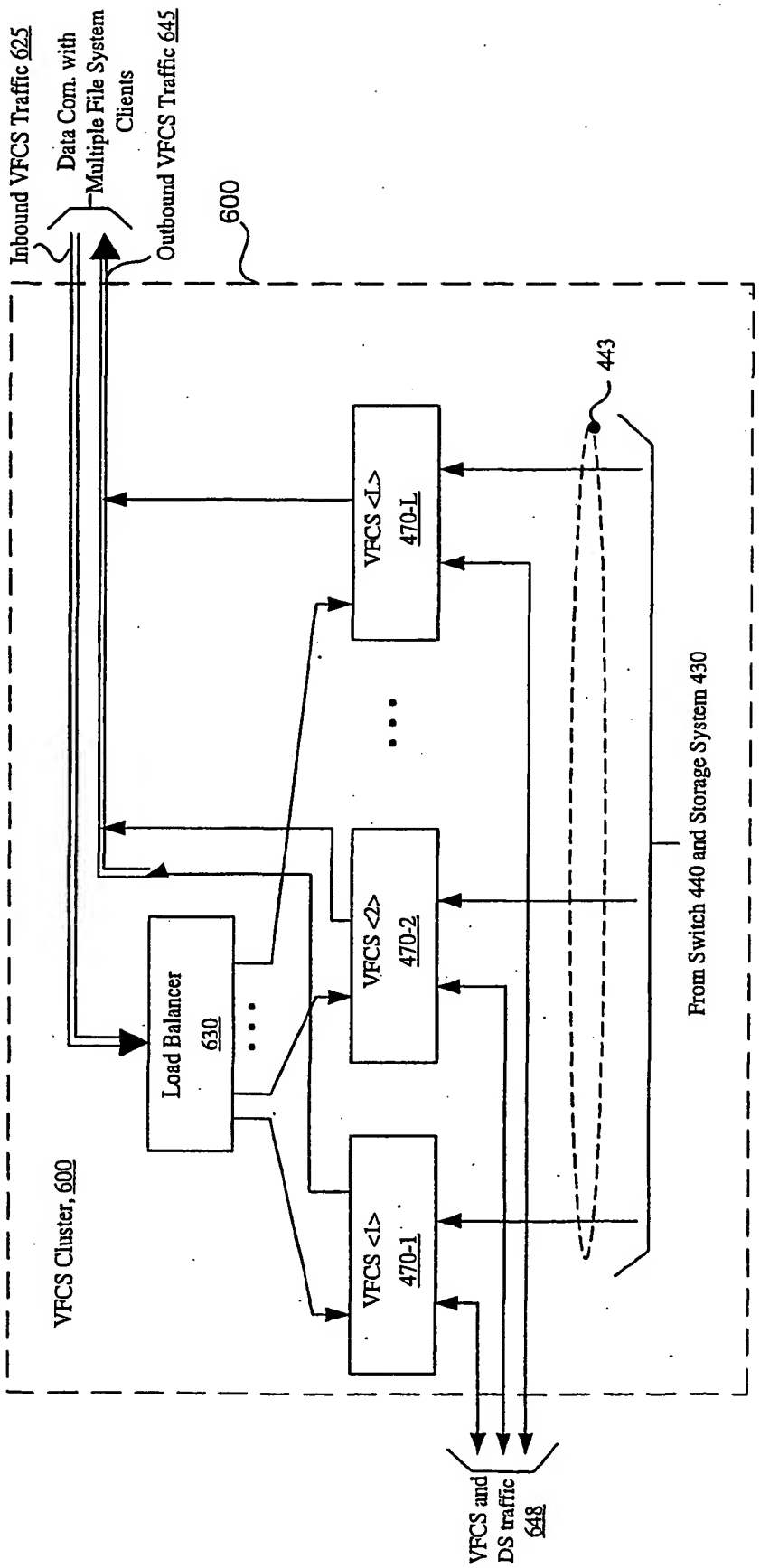

Inbound VFCS Traffic 625
Data Com. with Multiple File System Clients
Outbound VFCS Traffic 645
600
443
VFCS <L> 470-L
VFCS <2> 470-2
VFCS <1> 470-1
Load Balancer 630
VFCS Cluster, 600
From Switch 440 and Storage System 430
VFCS and DS traffic 648

Figure 6